

Exercise walk-through

7.4

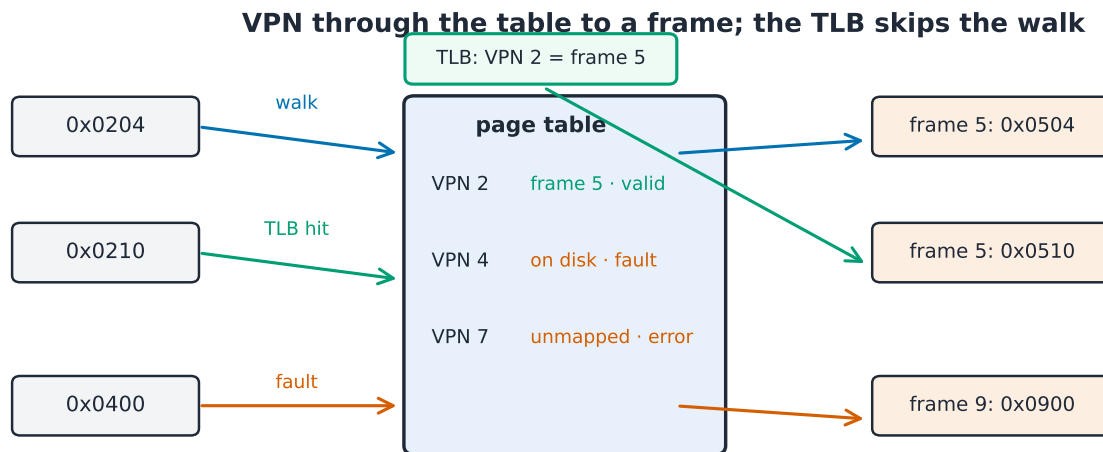
Virtual memory

University Computer Science

Virtual memory: walk, hit, fault

- 1 **Space** 16-bit virtual, 256-byte pages: 256 pages over 16 frames - each **process** sees 64 KiB while 4 KiB exists.
- 2 **Walk** 0x0204 is offset 4, VPN 2, frame 5 - physical 0x0504; 0x0210 hits the **TLB** at 0x0510, no walk.
- 3 **Fault** Invalid entry: block, evict (LRU, dirty first), read in, restart - 1 ns hit, 10 ns walk, 5 ms fault.

Page-table mapping



16-bit virtual, 256-byte pages: 256 pages; 12-bit physical: 16 frames

VPN through the table to a frame; the **TLB** skips the walk.

Exercises 2.1-2.3

2.1 20-bit virtual addresses with 1 KiB pages: how many virtual pages, and how many VPN bits? [3]

2.2 VPN 7 maps to frame 3. Translate `0x0705` (256-byte pages). [3]

2.3 Distinguish an invalid access from a page fault in one line. [3]

Exercises 2.1-2.3

2.1 20-bit virtual addresses with 1 KiB pages: how many virtual pages, and how many VPN bits? [3]

2.1

- $2^{20}/2^{10} = 2^{10} = 1024$ pages; VPN = 10 bits ✓.

2.2 VPN 7 maps to frame 3. Translate 0x0705 (256-byte pages). [3]

2.2

- Offset 0x05; physical = $3 \times 256 + 5 = 0x305$ ✓.

2.3 Distinguish an invalid access from a page fault in one line. [3]

2.3

- Invalid = the page was never mapped (error); a fault = mapped but not resident (fetch it) ✓.

Exercise 3.1

3.1 A machine has 24-bit virtual addresses, 2 KiB pages, and 1 MiB of physical memory.

Compute the page count, the VPN bits, the frame count and the frame bits, and the page-table entry count. Translate virtual $0x123456$ (offset, VPN). Compute the page-table size in bits if each entry holds frame bits + valid + dirty + reference. [8]

Exercise 3.1

3.1 A machine has 24-bit virtual addresses, 2 KiB pages, and 1 MiB of physical memory.

Compute the page count, the VPN bits, the frame count and the frame bits, and the page-table entry count. Translate virtual $0x123456$ (offset, VPN). Compute the page-table size in bits if each entry holds frame bits + valid + dirty + reference. [8]

3.1

- Pages: $2^{24}/2^{11} = 2^{13} = 8192$ ✓.
- Frames: $1 \text{ MiB}/2 \text{ KiB} = 512$ – frame bits 9; VPN bits $= 24 - 11 = 13$ ✓.
- Entries: one per page $= 8192$ ✓.
- Entry size $= 9 + 3 = 12$ bits; table $= 8192 \times 12 = 98\,304$ bits ✓.
- $0x123456$: offset $= 0x456$ (low 11 bits); VPN $= 0x123456 \gg 11 = 0x246$ ✓.

Exercise 3.2

3.2 A process's accesses hit the TLB 98% of the time; a miss walks the table in 20 ns; 1% of walks fault, each costing 4 ms.

Compute the average access time (TLB hit 1 ns). Recompute with faults at 0.1% of walks. State which term dominates each answer and the one-line design lesson. [7]

Exercise 3.2

3.2 A process's accesses hit the TLB 98% of the time; a miss walks the table in 20 ns; 1% of walks fault, each costing 4 ms.

Compute the average access time (TLB hit 1 ns). Recompute with faults at 0.1% of walks. State which term dominates each answer and the one-line design lesson. [7]

3.2

- Walk time per miss: 20 ns; faults add $0.01 \times 4 \text{ ms} = 40 \text{ } \mu\text{s}$ per walk ✓.
- $\text{AMAT} = 0.98 \times 1 + 0.02 \times (20 + 40\,000) \approx 801 \text{ ns}$ ✓.
- At 0.1%: $0.02 \times (20 + 4000) \rightarrow \text{AMAT} = 0.98 + 80.4 \approx 81.4 \text{ ns}$ ✓.
- The fault term dominates both answers; lesson: the hierarchy tolerates rarity, not frequency – locality is the budget ✓.